IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of: Teik-Chung Tan Gregory William Smaus	§ § § §	Group Art Unit: 2183 Examiner: Fennema, Robert E.
Serial No. 10/729,331	§ § § § §	Atty. Dkt. No.: 5500-91700/TT5404
Filed: December 5, 2003	§ §	
For: Multiple Control Sequences per Row of Microcode ROM		

APPEAL BRIEF

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir/Madam:

Further to the Notice of Appeal filed January 3, 2007, Appellants present this Appeal Brief. Appellants respectfully request that the Board of Patent Appeals and Interferences consider this appeal.

I. REAL PARTY IN INTEREST

As evidenced by the assignment recorded at Reel/Frame 014772/0976, the subject application is owned by Advanced Micro Devices, Inc., a corporation organized and existing under and by virtue of the laws of the State of Delaware, and now having its principal place of business at One AMD Place, P.O. Box 3453, Sunnyvale, CA 94088.

II. RELATED APPEALS AND INTERFERENCES

No other appeals, interferences or judicial proceedings are known which would be related to, directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

III. STATUS OF CLAIMS

Claims 1-27 are pending and stand finally rejected. The rejection of claims 1-27 is being appealed. A copy of claims 1-27 is included in the Claims Appendix herein below.

IV. STATUS OF AMENDMENTS

No amendments have been submitted subsequent to the final rejection.

V. SUMMARY OF CLAIMED SUBJECT MATTER

Independent claim 1 is directed to a microprocessor. (*See, e.g.,* FIG. 2, element 100; and p. 9, lines 3-5). The microprocessor includes a microcode ROM and a control sequence logic unit. (*See, e.g.,* FIG. 2, elements 155 and 152; and p. 9, lines 7-8).

A row in the microcode ROM stores a plurality of groups of microcode operations, where a group of the plurality of groups of microcode operations is part of a microcode routine. (*See, e.g.,* FIG. 3, elements 220A – 220X, especially 220X; p. 14, line 31 – p. 15, line 3; and p. 16, lines 1-3.) The row also stores an associated control sequence for each of the plurality of groups of microcode operations. (*See, e.g.,* FIGs. 3 and 4, elements 210A-210X; and p. 4, lines 6-7.)

The control sequence logic unit is coupled to the microcode ROM. (*See, e.g.,* FIG. 3, in which element 152 is coupled to element 155; and p. 4, lines 5-6). In response to accessing the group of microcode operations included in the microcode routine, the control sequence logic unit is configured to use the control sequence associated with the group of microcode operations to identify another row storing one or more next groups of microcode operations comprised in the microcode routine. (*See, e.g.*, p. 14, line 30 – p. 15, line 17.)

Independent claim 12 is directed to a computer system including a system memory and a microprocessor coupled to the system memory. (*See, e.g.,* FIG. 6, elements 100 and 404; FIG. 2, elements 100 and 200; p. 9, lines 3-12; and p. 20, lines 9-16.) The microprocessor contains a microcode ROM and control sequence unit similar to those described above with regard to claim 1. A row in the microcode ROM stores a plurality of groups of microcode operations, where a group of the plurality of groups of microcode operations is part of a microcode routine. (*See, e.g.,* FIG. 3, elements 220A – 220X, especially 220X; p. 14, line 31 – p. 15, line 3; and p. 16, lines 1-3.) The row also

stores an associated control sequence for each of the plurality of groups of microcode operations. (*See, e.g.,* FIGs. 3 and 4, elements 210A-210X; and p. 4, lines 6-8.)

The control sequence logic unit is coupled to the microcode ROM. (*See, e.g.,* FIG. 3, in which element 152 is coupled to element 155; and p. 4, lines 5-6). In response to accessing the group of microcode operations included in the microcode routine, the control sequence logic unit is configured to use the control sequence associated with the group of microcode operations to identify another row storing one or more next groups of microcode operations comprised in the microcode routine. (*See, e.g.*, p. 14, line 30 – p. 15, line 17.)

Independent claim 23 is directed to a method that includes storing a plurality of groups of microcode operations and a plurality of control sequences in a row in a microcode ROM, where each of the plurality of control sequences is associated with a respective one of the groups of microcode operations. (*See, e.g.,* FIG. 5, element 501; p. 4, lines 6-8; and p. 19, line 2-4.)

The method also includes, in response to accessing one of the plurality of groups of microcode operations, using the one of the plurality of control sequences associated with that one of the plurality of groups to identify a next group of microcode operations to output from the microcode ROM. (*See, e.g.,* FIG. 5, elements 502 and 503; p. 14, line 30 – p. 15, line 17; and p. 19, lines 4-12.)

Independent claim 27 is directed to a system similar to that of claim 12 and that includes a microcode ROM similar to that described above regarding claims 1 and 12. (*See, e.g.,* FIGs. 2 and 6, processor 100, which includes a microcode ROM 155; p. 9, lines 3-8; and p. 20, lines 9-12.) A row in the microcode ROM stores a plurality of groups of microcode operations and an associated control sequence for each of the plurality of groups. (*See, e.g.,* FIG. 3, elements 220A – 220X, especially 220X; p. 14,

line 31 – p. 15, line 3; and p. 16, lines 1-3.)

The system also includes means for accessing a control sequence associated with one of the plurality of groups of microcode operations and responsively accessing a next group of microcode operations stored in the microcode ROM. (*See, e.g.,* FIGs. 2 and 3, control sequence unit 152; p. 14, line 30 – p. 15, line 17; and p. 15, lines 14-17.)

The summary above describes various examples and embodiments of the claimed subject matter; however, the claims are not necessarily limited to any of these examples and embodiments. The claims should be interpreted based on the wording of the respective claims.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Claims 1-3, 5-10, 12-14, 16-21, 23, 24 and 27 stand finally rejected under 35 U.S.C. § 102(b) as being anticipated by Tredennick et al. (U.S. Patent 4,338,661) (hereinafter "Tredennick").

2. Claims 4, 11, 15, 22, and 25-26 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Tredennick in view of Yoshida (U.S. Patent 5,761,470).

VII. ARGUMENT

First ground of rejection:

Claims 1-3, 5-10, 12-14, 16-21, 23, 24 and 27 stand finally rejected under 35 U.S.C. § 102(b) as being anticipated by Tredennick et al. (U.S. Patent 4,338,661) (hereinafter "Tredennick"). Appellants traverse this rejection for at least the following reasons. Different groups of claims are addressed under their respective subheadings.

Claims 1, 2, 12, 13, and 23:

Regarding independent claim 1, contrary to the Examiner's assertion, Tredennick clearly fails to teach or suggest *a microcode ROM, wherein a row in the microcode ROM stores a plurality of groups of microcode operations, wherein a group of the plurality of groups of microcode operations is comprised in a microcode routine, and wherein the row stores an associated control sequence for each of the plurality of groups of microcode operations.*

In the Advisory Action, the Examiner submits that Tredennick teaches *wherein a row in the microcode ROM stores a plurality of groups of microcode operations*, using an example in which one row in the nano control of Tredennick includes four nanowords: swap1, swap2, tasm1, and tasm2. The Examiner interprets this example as teaching a plurality of groups of microcode operations (swap and tasm, each with two parts). The Examiner further asserts, "the groups could contain a single instruction, with some rows containing up to 4 groups of operation, based on the particular row and exact length of ROM chosen to implement the design." However, a single instruction is not "a group of microcode operations (plural) … comprised in a microcode routine" as recited in claim 1, and having the additional limitations of claim 1 discussed below. The plain wording of claim 1 requires that the group referred to in subsequent limitations contain multiple operations. For example, claim 1 recites, in part, "wherein in response to accessing the group of microcode operations comprised in the microcode routine..." and later, "the

control sequence associated with the group of microcode operations…" These limitations are clearly not anticipated by a group containing only one microcode operation, as the Examiner suggests.

Appellants also assert that Tredennick clearly fails to teach or suggest *wherein the row stores an associated control sequence for each of the plurality of groups of microcode operations*. The Examiner cites column 15, lines 52-55 and 59-66 as teaching this limitation. These passages describe two formats for microwords. In a microword of a conditional branch type (type II), bits 7 thru 14 comprise a next micro ROM base address (NMBA) for the micro and nano control stores, and are augmented by 2 additional bits supplied by branch control logic (C0 and C1) in order to specify the next address for the control stores (i.e., in order to select a single word line for the micro and nano control stores). In microwords having format type I, bits 2 and 3 comprise a type field (TY) which specifies the source of the next address for the control stores as being from one of the 3 possible addresses provided by the instruction register sequence decoder or from a direct branch address provided by bits 5 thru 14 of the microword. Thus, these passages disclose that information stored in each microword may be used in determining the next address used to select a single word line (and, therefore, a single next operation) for each of the control stores. They clearly do not describe a control sequence being **associated with a group of *multiple* instructions** comprised in a row in the microcode, as in Appellants' claimed invention. There is nothing in these passages or elsewhere that teaches or discloses a row storing an associated control sequence for each of the plurality of groups of microcode operations stored in the row.

Using the Examiner's example in the eighth row of FIG. 11A, there is no control sequence in Tredennick associated with a group of operations that includes swap1 and swap2, nor with a group of operations that includes tasm1 and tasm2. Instead, each of the individual microwords corresponding to these nanowords includes a micro ROM base address for the next individual microword and nanoword to be accessed.

Further regarding claim 1, Tredennick fails to teach or suggest a control sequence logic unit coupled to the microcode ROM, wherein in response to accessing the group of microcode operations comprised in the microcode routine, the control sequence logic unit is configured to use the control sequence associated with the group of microcode operations to identify another row storing one or more next groups of microcode operations comprised in the microcode routine. The Examiner cited column 15, lines 37-40 as teaching this limitation ("it selects the next line of the ROM, which is output from the microcode ROMs as shown in column 15, lines 52-55 and 59-66.") First, as discussed above, Tredennick does not disclose a control sequence **associated with a group of microcode operations**. In addition, column 15, lines 37-40 merely describes that the micro ROM is addressed by the 10-bit output of address selection block 64. Furthermore, the Examiner's statement "it selects the next line of the ROM" has no basis in Tredennick. As discussed above, column 15, lines 52-55 and 59-66 describes how the next address to be decoded for the control stores is determined for microwords having type I and type II formats. None of these descriptions includes a control sequence identifying an other row storing one or more next groups of microcode operations comprised in the (same) microcode routine. Instead, for each individual operation, a next address is provided for a next individual operation.

Using the Examiner's example, there is clearly not a control sequence (or next micro ROM address) associated with a group of operations including swap1 and swap2 that identifies an other row storing one or more next groups of microcode operations comprised in the (same) microcode routine, as the "swap1" microcode routine includes only the two operations swap1 and swap2 on the eighth row of FIG. 11A (see Appendix A). Similarly, there is clearly not a control sequence (or next micro ROM address) associated with a group of operations containing trap1 and trap2 that identifies an other row storing one or more next groups of microcode operations comprised in the (same) microcode routine, as the "trap1" microcode routine includes only the two operations trap1 and trap2 on the eighth row of FIG 11A (see Appendix A). Appellants assert that nothing in Tredennick teaches a control sequence associated with a group microcode operations comprised in a microcode routine that is used to identify an other row storing

one or more next groups of microcode operations comprised in the (same) microcode routine, as required by claim 1.

For at least the reasons above, Appellants again assert that Tredennick clearly does not teach all of the limitations of Appellants' claim 1. Therefore, the rejection of claim 1 is unsupported by the cited art and removal thereof is respectfully requested.

Independent claims 12 and 23 include limitations similar to claim 1, and so the arguments presented above apply with equal force to these claims, as well.

Claim 27:

Regarding independent claim 27, contrary to the Examiner's assertion, Tredennick clearly fails to teach or suggest *a microcode ROM, wherein a row in the microcode ROM stores a plurality of groups of microcode operations and wherein the row stores an associated control sequence for each of the plurality of groups.* In the previous Office Action, the Examiner again cited column 15, lines 33-34, 36-37, 52-55, and 59-66 as teaching these limitations. However, as discussed above regarding claim 1, Tredennick does not teach or suggest a row in the microcode storing groups of microcode operations and a control sequence associated with each of the groups.

Further regarding claim 27, Tredennick fails to teach or suggest *means for accessing a control sequence associated with one of the plurality of groups of microcode operations and responsively accessing a next group of microcode operations stored in the microcode ROM.* The Examiner again cited "Column 15, lines 37-40, it selects the next line of the ROM, which is output from the microcode ROMs as shown in column 15, lines 52-55 and 59-66." However, as discussed above regarding claim 1, Tredennick does not teach or suggest groups of microcode operations stored in a row that also stores a control sequence associated with one of the **groups** of microcode operations, and therefore, cannot teach or suggest accessing such a control sequence or responsively

accessing a next **group** of microcode operations stored in the microcode ROM. Therefore, Tredennick cannot be said to anticipate claim 27.

In the Response to Arguments section of the Final Action, the Examiner submits that claim 27, "has similar limitations, except the control addresses a group instead of a row, which is taught by Tredennick as explained above." Appellants assert that, as discussed above regarding claim 1 and claim 27, Tredennick does not teach a control sequence associated with each one of a plurality of groups of microcode operations, nor that such control sequences are stored in a row containing each of the plurality of groups of microcode operations, as required by claim 27.

For at least the reasons above, the rejection of claim 27 is unsupported by the cited art and removal thereof is respectfully requested.

Claims 3 and 14:

Regarding claim 3, contrary to the Examiner's assertion, Tredennick clearly fails to teach or suggest *wherein the control sequence logic unit is configured to identify which of a plurality of groups of microcode operations stored in the other row of the microcode ROM are comprised in the microcode routine based on information contained in the control sequence associated with the group of microcode operations stored in the row*. The Examiner cited column 15, lines 52-55 and 59-66, "which defines which row and position the next group is located," as teaching these limitations. However, this citation does not teach identifying a row and position for a next group of microcode operations. Instead, this passage describes how the next address is determined for an individual microword having a type I or type II format. There is nothing in the Examiner's citation or elsewhere in Tredennick that teaches or suggests that identifying the next address (or "a row and position") has anything to do with identifying which of a plurality of groups of microcode operations stored in the other row of the microcode ROM are comprised in the (same) microcode routine, as required by claim 3, only that it identifies a next (individual) operation.

In the Response to Arguments section of the Final Action, the Examiner submits, "The cited portion discloses that the address directs the processor to the next group in the routine, which must specify a row, as discussed in Claim 1. It can be further seen in Figure 11a that the 10-bit address does select a position using bits A1 and A0." Appellants again assert that the address selects a single microword or nanoword, not a group of microcode operations stored in a row and comprised in the microcode routine.

For at least the reasons above, the rejection of claim 3 is unsupported by the cited art and removal thereof is respectfully requested.

Claim 14 includes limitations similar to claim 3, and so the arguments presented above apply with equal force to this claim, as well.

Claims 5, 16, and 24:

Regarding claim 5, contrary to the Examiner's assertion, Tredennick clearly fails to teach or suggest *wherein if the group of microcode operations comprises at least one branch operation, the control sequence logic unit is configured to identify the next group of microcode operations in the microcode routine dependent on a branch prediction as well as the control sequence associated with the group of microcode operations*. The Examiner cited column 15, lines 49-55 ("teach a microcode instruction for branches") and column 17, lines 29-32 ("show that the outcome either way will be in the same row specified by the control sequence") as teaching these limitations. The Examiner's citation in column 17 states, "Thus, two microwords which serve as alternate destinations for a particular conditional branch type microword must be placed in the same logical row of the micro ROM." Appellants assert that this citation does not describe identifying the next group of microcode operations in a microcode routine, but instead describes two alternate (individual) addresses that may be the destination of a branch operation. Furthermore, the Examiner's remarks appear to teach away from identifying the next group of operations dependent on a branch prediction. Instead, they imply that no branch

prediction is necessary (or performed) because "the outcome either way will be in the same row." **In fact, branch prediction is not disclosed in Tredennick.** Therefore, Tredennick clearly cannot be said to anticipate claim 5.

In the Response to Arguments section of the Final Action, the Examiner submitted, "Applicant has argued that Tredennick teaches away from the claims because both outcomes are located in the same row. However, Tredennick teaches this is done to minimize the size of storage, to prevent multiple instances of the branch from being put in memory (Column 2, Lines 40-50), and as explained above, Tredennick can specific positions inside a row, the fact that they are in the same row is space optimization." The Examiner has misunderstood Appellants' argument. Appellants' argument was that Tredennick taught away from this limitation for several reasons. One is that all of the possible next microwords to be accessed will be in the same row, therefore the next row to be accessed is already known. The fact that the outcomes are positioned in the same row to save space is irrelevant. The second is that no branch prediction is described, as required by Appellants' claim. In addition, Tredennick does not teach identifying a row containing a next **group** of microcode operations comprised in the microcode routine, as required by Appellants' claim, only a next individual microword (and/or nanoword).

For at least the reasons above, the rejection of claim 5 is unsupported by the cited art and removal thereof is respectfully requested.

Claims 16 and 24 include limitations similar to claim 5, and so the arguments presented above apply with equal force to these claims, as well.

Claims 6 and 17:

Regarding claim 6, contrary to the Examiner's assertion, Tredennick clearly fails to teach or suggest *wherein the microcode ROM is divided into a plurality of segments, wherein a same number of groups of microcode operations is stored in each row of a given one of the plurality of segments, and wherein each row in the given one of the*

plurality of segments stores a different number of groups of microcode operations than each row in each other one of the plurality of segments. The Examiner cited column 19, lines 22-27 as teaching this limitation ("for each row, the address may represent one, two, four, or up to eight different groups. So there are segments in the sense that some lines can contain a different number of groups than the other lines.") The Examiner has misquoted column 19. The Examiner's citation states, "Each word line in the micro ROM is represented by only one input address. Each word line in the nano ROM however may represent one, two, or four possible different input addresses. In the preferred embodiment of the data processor, a word line in the nano ROM may represent as many as eight different input addresses." However, "a word line" in Tredennick is not a row in the microcode ROM that stores a plurality of groups of microcode operations, as in Appellants' claimed invention. Instead, "a word line" selects a single microword and/or a single nanoword from the micro ROM and nano ROM, respectively. (See, e.g., column 19, lines 13-22: "the same address is presented to the decoders of both the micro ROM and the nano ROM. For any input address, there will be no more than one word line in each ROM which remains high. The line which remains high will cause the appropriate output value to be generated as the micro ROM output word and the nano ROM output word according to the coding at the intersection of the selected word line and the output columns." There is nothing in Tredennick that teaches or suggest the microcode ROM is divided into a plurality of segments having the limitations recited in claim 6 ("wherein a same number of groups of microcode operations is stored in each row of a given one of the plurality of segments, and wherein each row in the given one of the plurality of segments stores a different number of groups of microcode operations than each row in each other one of the plurality of segments.") Therefore, Tredennick cannot be said to anticipate claim 6.

In the Response to Arguments section of the Final Action, the Examiner submitted that "in Tredennick, the word line in one ROM is a row (the micro ROM), or a "group" in the nano ROM. Thus, when reading Tredennick as it was intended, a "row" or "word" in the micro ROM corresponds to multiple "words" in the nano ROM." **The Examiner is incorrect**. Tredennick teaches that one input address may be used to

identify one word (which is not a "row" as defined by Appellants' claims) in the micro ROM and one word in the nano ROM. The mapping of multiple nano words to one micro word described by the Examiner is incorrect. Instead Tredennick teaches that more than one input address may identify the same nano word, as in the example detailed in the previous Response.

For at least the reasons above, the rejection of claim 6 is unsupported by the cited art and removal thereof is respectfully requested.

Claim 17 includes limitations similar to claim 6, and so the arguments presented above apply with equal force to this claim, as well.

Claims 7-9 and 18-20:

For reasons similar to those discussed above regarding claim 6, Tredennick fails to teach or suggest the limitations of claims 7-9. The Examiner cited column 19, lines 22-27 as teaching these limitations by the same reasoning applied to his rejection of claim 6. However, as discussed above, this citation has nothing to do with a plurality of segments in a microcode ROM, much less with such segments having the limitations recited in claims 7-9. As discussed above regarding claims 1 and 6, Tredennick clearly does not teach anything about segments of microcode operations defined or stored as in Appellants' claimed invention. Furthermore, Appellants again assert that Tredennick teaches nothing about the maximum width of any such segments or what would be stored in such a segment (e.g., *one group of microcode operations and one associated control sequence per row*, as claimed.) Therefore, Tredennick cannot be said to anticipate these claims. For at least the reasons above, the rejection of claims 7-9 is unsupported by the cited art and removal thereof is respectfully requested. Claims 18-20 include limitations similar to claims 7-9 and so the arguments presented above apply with equal force to these claims, as well.

Claims 10 and 21:

Regarding claim 10, contrary to the Examiner's assertion, Tredennick clearly fails to teach or suggest *wherein the control sequence logic unit is configured to identify a position of one or more groups of microcode operations and a position of one or more control sequences dependent on which of the plurality of segments of the microcode ROM stores the one or more groups of microcode operations*. The Examiner cites column 15, lines 52-55 and 59-66, "which defines which row and position the next group is located," as teaching this limitation. However, as discussed above, this citation this citation does not teach identifying a row and position for a next group of microcode operations. Instead, this passage describes how the next address (which is clearly not the same as "a row and position") is determined for an individual microword having a type I or type II format. As discussed above, Tredennick does not teach or suggest a microcode ROM organized according to the rows, routines, groups, and segments of Appellants' claimed invention, and clearly does not teach identifying a position of one or more groups of microcode operations and control sequences dependent on which of a plurality of segments stores the one or more groups of microcode operations. Therefore, Tredennick clearly does not anticipate claim 10.

For at least the reasons above, the rejection of claim 10 is unsupported by the cited art and removal thereof is respectfully requested.

Claim 21 includes limitations similar to claim 10, and so the arguments presented above apply with equal force to this claim, as well.

Second ground of rejection:

Claims 4, 11, 15, 22, and 25-26 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Tredennick in view of Yoshida (U.S. Patent 5,761,470).

Appellants traverse this rejection for at least the following reasons. Different groups of claims are addressed under their respective subheadings.

Claims 4, 15, and 25:

Regarding claim 4, contrary to the Examiner's assertion, Tredennick in view of Yoshida clearly fails to teach or suggest *wherein if fewer than all of a plurality of groups of microcode operations stored in the other row of the microcode ROM are comprised in the microcode routine, the control sequence logic unit is configured to substitute NOPs for the microcode operations comprised in the groups not comprised in the microcode routine when outputting the row to the scheduler*. The Examiner submits that Tredennick teaches the microprocessor of claim 3, which Appellant traverses above. The Examiner admits that Tredennick fails to teach the above-referenced limitation of claim 4, and relies on Yoshida to teach it.

Yoshida teaches of a VLIW machine, which exploits parallelism by executing multiple instructions simultaneously. Yoshida teaches that if the conventional VLIW machine cannot execute an instruction from the word in parallel, it inserts a NOP in its place, as it has to execute some instruction (column 1, lines 51-56). The Examiner submits that, "Given the advantage of higher speed though parallelism, one of ordinary skill in the art at the time the invention was made would have converted Tredennick's invention to operate in a parallel fashion such as a VLIW machine to increase the speed and performance."

Appellants argue that converting Tredennick's invention to operate in a parallel fashion would clearly (and dramatically) change the principle of operation of his invention. In the Response to Arguments section of the Final Action, the Examiner disagrees, stating, "Tredennick's invention is essentially a machine which executes instructions using microcode stored in a ROM. Modifying the invention to work in a parallel manner does not affect this principle in any way, shape, or form. While the modification to make parallel may change certain elements of the invention, the overall

principle of the invention will remain the same, to execute instructions. Therefore, the inclusion of Yoshida is proper, as the principle of the invention has not changed in any way." Appellants disagree. The Examiner seems to imply that there is no difference in the operation of Tredennick's invention and Yoshida's invention other than changes to certain elements of the invention due to Yoshida's parallelism. This is clearly not the case. Appellants assert that many elements of Yoshida's invention would be greatly affected by such a modification and that each of these elements would necessarily operate differently than similar elements of Tredennick's invention to support parallelism (e.g., branch control logic, instruction sequencing logic, instruction fetch logic, instruction decoding logic, just to name a few). For example, even the formats of instructions decoded in Yoshida's invention are vastly different than those of Tredennick's invention since Yoshida's invention uses Very Long Instruction Word, or VLIW, instructions, each of which specifies a plurality of operations by one instruction. Appellants again assert that such a complete change in both the operation and structure of so many elements of the invention of Tredennick would clearly change the principle of operation of Tredennick and would, therefore, not be obvious to one of ordinary skill in the art at the time the invention was made.

Appellants also argue that converting Tredennick's invention "to operate in parallel fashion" would not necessarily result in Appellants' claimed invention. The feature described in Yoshida "if the conventional VLIW machine cannot execute an instruction from the word in parallel, it inserts a NOP in its place" (emphasis added) refers to a VLIW processor that consumes one instruction even when there are no operations which can be executed in parallel. In this case, a number of operation fields specifying null operations (No Operation: NOP) are generated, and the amount of instruction code becomes very big. This is clearly not the same as the limitation in Appellants' claim 4, which recites "*wherein if fewer than all of a plurality of groups of microcode operations stored in the other row of the microcode ROM are comprised in the microcode routine, the control sequence logic unit is configured to substitute NOPs for the microcode operations comprised in the groups not comprised in the microcode routine when outputting the row to the scheduler,*" as Yoshida does not describe the

organization of a microcode ROM at all, much less one in which groups of microcode operations are stored in particular rows. Furthermore, it is not clear that all parallel processors necessarily include this "feature" of Yoshida, nor does the reference (or the Examiner, in his remarks) explain how this feature would be implemented in Tredennick's processor if "converted to operate in parallel fashion."

In the Response to Arguments section of the Final Action, the Examiner submits, "Yoshida does not need to teach the structural elements of the claim, Tredennick is used for that purpose. Yoshida is used to show that one of ordinary skill in the art would have recognized the need to insert NOP's in place of instructions fed to the machine that could not be executed in order to ensure correct execution, which is a motivation for doing so to one of ordinary skill in the art." Appellants disagree. The Examiner seems to be arguing that Tredennick could not operate correctly without this feature if it were converted to operate in parallel fashion. However, as noted above, the Examiner has not explained how he thinks Tredennick could be so converted or why he believes this feature would be necessary after such a conversion. Appellants again assert that this is not a necessary feature of all microprocessors that operate in parallel fashion. The referenced feature of Yoshida solves a problem that may or may not be present if Tredennick were converted to operate in parallel fashion. Furthermore, the Examiner's statement that Yoshida teaches "the need to insert NOP's in place of instructions fed to the machine that could not be executed" is incorrect. Instead, Yoshida teaches inserting NOP's where no operations are available to be fed to one of the machines (i.e., when there are no available operations that can be executed in parallel.) **Applicants again assert that this feature does not teach the limitations of claim 4, which recites substituting NOPs for microcode operations of another microcode routine when a row is output to the scheduler.** Yoshida clearly does not teach substituting NOPs for microcode operations so that they will not be output to a scheduler, nor would including Yoshida's feature (feeding NOPs to one data processor with nothing to execute when another data processor executing in parallel does have an operation to execute) in a modified invention of Tredennick teach this limitation.

For at least the reasons above, Appellants assert that Tredennick in view of Yishida fails to teach or suggest all the limitations of claim 4.

For at least the reasons above, the rejection of claim 4 is unsupported by the cited art and removal thereof is respectfully requested.

Claims 15 and 25 include limitations similar to claim 4, and so the arguments presented above apply with equal force to these claims, as well.

Claims 11, 22, and 26:

For reasons similar to those discussed above regarding claim 4, Tredennick in view of Yoshida clearly fails to teach or suggest *wherein a plurality of groups of microcode operations stored in the other row of the microcode ROM are comprised in the microcode routine and are output during a single access*, as recited in claim 11. The Examiner admits that Tredennick fails to teach this limitation, and again relies on Yoshida's VLIW machine to teach this limitation. As discussed above, the Examiner's proposed modification of the prior art would change the principle of operation of the prior art invention being modified, and thus the teachings of the references are not sufficient to render the claims *prima facie* obvious.

In addition, the Examiner's citations in Yoshida do not teach *a plurality of groups of microcode operations stored in the other row of the microcode ROM are comprised in the microcode routine and are output during a single access*. They teach that one VLIW word can specify a plurality of instructions (column 1, lines, 25-30), but they do not disclose that instructions are output during a single access, that they are stored in rows, or wherein a plurality of groups of microcode operations stored in the other row of the microcode ROM are comprised in a microcode routine.

In the Response to Arguments section of the Final Action, the Examiner further submits, "Yoshida does not need to teach instructions stored in rows in a ROM, and the

other structural elements of the claim, that is what Tredennick was used to teach. Yoshida provides motivation for accessing multiple groups of the groups in the row that Tredennick teaches, to increase performance by being able to potentially execute multiple operations at once, and that one of ordinary skill in the art would have been able to modify Tredennick to do so." Appellants disagree. First, Yoshida is able to achieve parallelism only by teaching a Very Long Instruction Word that includes multiple operations that may be executed in parallel. Yoshida cannot suggest accessing multiple groups of the groups in the row of Tredennick because Yoshida does not teach accesses multiple groups of groups of operations itself or multiples ones of the VLIW instructions. Instead it teaches accessing a single instruction (which may be said to include a single group of operations) at a time. Appellants again assert that the modification of Tredennick to operate in parallel fashion would completely change both the structure and operation of the invention and that even if it were combined with the referenced features of Yoshida, it would not teach the limitations of claim 11.

For at least the reasons above, the rejection of claim 11 is unsupported by the cited art and removal thereof is respectfully requested.

Claims 22 and 26 include limitations similar to claim 11, and so the arguments presented above apply with equal force to these claims, as well.

CONCLUSION

For the foregoing reasons, it is submitted that the Examiner's rejection of claims 1-27 was erroneous, and reversal of his decision is respectfully requested.

The Commissioner is authorized to charge the appeal brief fee of $500.00 and any other fees that may be due to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5500-91700/RCK.

Respectfully submitted,

/Robert C. Kowert/
Robert C. Kowert, Reg. #39,255
Attorney for Appellants

Meyertons, Hood, Kivlin, Kowert & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
(512) 853-8850

Date: March 5, 2007

VIII. CLAIMS APPENDIX

The claims on appeal are as follows.

1. A microprocessor, comprising:

a microcode ROM, wherein a row in the microcode ROM stores a plurality of groups of microcode operations, wherein a group of the plurality of groups of microcode operations is comprised in a microcode routine, and wherein the row stores an associated control sequence for each of the plurality of groups of microcode operations; and

a control sequence logic unit coupled to the microcode ROM, wherein in response to accessing the group of microcode operations comprised in the microcode routine, the control sequence logic unit is configured to use the control sequence associated with the group of microcode operations to identify an other row storing one or more next groups of microcode operations comprised in the microcode routine.

2. The microprocessor of claim 1, wherein at least one of the plurality of groups of microcode operations stored in the row is part of a different microcode routine.

3. The microprocessor of claim 1, wherein the control sequence logic unit is configured to identify which of a plurality of groups of microcode operations stored in the other row of the microcode ROM are comprised in the microcode routine based on information contained in the control sequence associated with the group of microcode operations stored in the row.

4. The microprocessor of claim 3, wherein if fewer than all of a plurality of groups of microcode operations stored in the other row of the microcode ROM are comprised in the microcode routine, the control sequence logic unit is configured to

substitute NOPs for the microcode operations comprised in the groups not comprised in the microcode routine when outputting the row to the scheduler.

5. The microprocessor of claim 1, wherein if the group of microcode operations comprises at least one branch operation, the control sequence logic unit is configured to identify the next group of microcode operations in the microcode routine dependent on a branch prediction as well as the control sequence associated with the group of microcode operations.

6. The microprocessor of claim 1, wherein the microcode ROM is divided into a plurality of segments, wherein a same number of groups of microcode operations is stored in each row of a given one of the plurality of segments, and wherein each row in the given one of the plurality of segments stores a different number of groups of microcode operations than each row in each other one of the plurality of segments.

7. The microprocessor of claim 6, wherein groups of microcode operations stored in a same one of the plurality of segments have a same maximum width.

8. The microprocessor of claim 7, wherein groups of microcode operations stored in one of the plurality of segments have a maximum width that is different from a maximum width of groups of microcode operations stored in another one of the plurality of segments.

9. The microprocessor of claim 8, wherein one of the plurality of segments stores one group of microcode operations and one associated control sequence per row.

10. The microprocessor of claim 6, wherein the control sequence logic unit is configured to identify a position of one or more groups of microcode operations and a position of one or more control sequences dependent on which of the plurality of segments of the microcode ROM stores the one or more groups of microcode operations.

11. The microprocessor of claim 1, wherein a plurality of groups of microcode operations stored in the other row of the microcode ROM are comprised in the microcode routine and are output during a single access.

12. A computer system, comprising:

a system memory; and

a microprocessor coupled to the system memory, comprising:

a microcode ROM, wherein a row in the microcode ROM stores a plurality of groups of microcode operations, wherein one of the plurality of groups of microcode operations is comprised in a particular microcode routine, and wherein the row stores an associated control sequence for each of the plurality of groups of microcode operations; and

a control sequence logic unit coupled to the microcode ROM, wherein in response to accessing the group of microcode operations comprised in the microcode routine, the control sequence logic unit is configured to use the control sequence associated with the group of microcode operations to identify an other row storing one or more next groups of microcode operations comprised in the microcode routine.

13. The computer system of claim 12, wherein at least one of the plurality of groups of microcode operations stored in the row is part of a different microcode routine.

14. The computer system of claim 12, wherein the control sequence logic unit is configured to identify which of a plurality of groups of microcode operations stored in the other row of the microcode ROM are comprised in the microcode routine, based on

information contained in the control sequence associated with the group of microcode operations stored in the row.

15. The computer system of claim 14, wherein if fewer than all of a plurality of groups of microcode operations stored in the other row of the microcode ROM are comprised in the microcode routine, the control sequence logic unit is configured to substitute NOPs for the microcode operations comprised in the groups not comprised in the microcode routine when outputting the row to the scheduler.

16. The computer system of claim 12, wherein if the group of microcode operations comprises at least one branch operation, the control sequence logic unit is configured to identify the next group of microcode operations in the microcode routine dependent on branch prediction as well as the control sequence associated with the group of microcode operations.

17. The computer system of claim 12, wherein the microcode ROM is divided into a plurality of segments, wherein a same number of groups of microcode operations is stored in each row of any of the plurality of microcode ROM segments, and wherein the number of groups of microcode operations stored in a row in one of the plurality of microcode ROM segments differs from the number of groups of microcode operations stored in a row in another one of the plurality of microcode ROM segments.

18. The computer system of claim 17, wherein groups of microcode operations stored in any one of the plurality of microcode ROM segments have a same maximum width.

19. The computer system of claim 18, wherein groups of microcode operations stored in one of the plurality of microcode ROM segments have a maximum width that is different from a maximum width of groups of microcode operations stored in another one of the plurality of microcode ROM segments.

20. The computer system of claim 19, wherein one of the plurality of microcode ROM segments stores one group of microcode operations and one associated control sequence per row.

21. The computer system of claim 17, wherein the control sequence logic unit is configured to identify a position of one or more groups of microcode operations within a row and their associated control sequences dependent on which of the plurality of segments of the microcode ROM stores the one or more groups of microcode operations.

22. The computer system of claim 12, wherein a plurality of groups of microcode operations stored in the other row of the microcode ROM and comprised in the microcode routine are output during a single access.

23. A method, comprising:

storing a plurality of groups of microcode operations and a plurality of control sequences in a row in a microcode ROM, wherein each of the plurality of control sequences is associated with a respective one of the groups of microcode operations; and

in response to accessing one of the plurality of groups of microcode operations, using the one of the plurality of control sequences associated with that one of the plurality of groups to identify a next group of microcode operations to output from the microcode ROM.

24. The method of claim 23, further comprising identifying the next group of microcode operations based on one or more branch predictions as well as the one of the plurality of control sequences if the one of the plurality of groups of microcode operations includes one or more branch operation.

25. The method of claim 23, further comprising substituting NOPs for one or more groups of microcode instructions stored in a same row as the next group of microcode operations dependent on the one of the plurality of control sequences.

26. The method of claim 25, wherein the groups of microcode operations comprised in the microcode routine and the NOPs are output as a single line.

27. A system, comprising:

a microcode ROM, wherein a row in the microcode ROM stores a plurality of groups of microcode operations and wherein the row stores an associated control sequence for each of the plurality of groups; and

means for accessing a control sequence associated with one of the plurality of groups of microcode operations and responsively accessing a next group of microcode operations stored in the microcode ROM.

IX. EVIDENCE APPENDIX

No evidence submitted under 37 CFR §§ 1.130, 1.131 or 1.132 or otherwise entered by the Examiner is relied upon in this appeal.

X. RELATED PROCEEDINGS APPENDIX

There are no related proceedings.